

## LIST OF CLAIMS

1. (Previously Presented) A method of processing data comprising:  
when a software buffer index points to a first buffer containing processed data, synchronizing said software buffer index to a hardware buffer index by sequentially searching through a plurality of buffers containing data to determine whether there is a second buffer with unprocessed data; and  
if there is said second buffer with unprocessed data, resetting said software buffer index to a next available buffer having processed data following said second buffer, and otherwise stopping said searching when each buffer of said plurality of buffers has been searched and a buffer with unprocessed data is not found.
2. (Previously Presented) The method of processing data as described in Claim 1, wherein said synchronizing further comprises:  
synchronizing said hardware buffer index and said software buffer index in response to an interrupt indicating data has been stored in one of said plurality of buffers and is ready for processing.
3. (Previously Presented) The method of processing data as described in Claim 1, wherein said synchronizing further comprises:  
ignoring a first interrupt indicating data has been stored in one of said plurality of buffers and is ready for processing when said software buffer index points to said first buffer containing processed data; and  
synchronizing said hardware buffer index and said software buffer index in response to a second interrupt indicating data has been stored in one of said

plurality of buffers and is ready for processing when said software buffer index points to said first buffer containing processed data for a second time.

4. (Original) The method of processing data as described in Claim 1, further comprising:

determining if said first buffer contains processed data; and  
processing data in said first buffer if said data is unprocessed.

5. (Previously Presented) The method of processing data as described in Claim 1, wherein said synchronizing further comprises:

wrapping around to a start buffer after searching the end buffer in said plurality of buffers when sequentially searching through said plurality of buffers, said plurality of buffers sequentially beginning with a start buffer and ending with an end buffer.

6. (Previously Presented) The method of processing data as described in Claim 1, further comprising:

when said software buffer index points to said first buffer containing processed data, using a value of said software buffer index corresponding to said first buffer as a reference value;

incrementing said software buffer index as each buffer of said plurality of buffers is searched, wherein when said software buffer index reaches one end of a range of possible values it is reset to the other end of said range; and

stopping said searching when said software buffer index reaches said reference value without finding a buffer in said plurality of buffers with unprocessed data.

7. (Original) The method of processing data as described in Claim 1, wherein each of said plurality of buffers is a local area network (LAN) buffer for storing LAN packets of data.

8. (Original) The method of processing data as described in Claim 7, wherein said software buffer index is a LAN software buffer index, and said hardware buffer index is a LAN hardware buffer index.

9. (Original) The method of processing data as described in Claim 1, further comprising:  
processing said unprocessed data in said second buffer.

10. (Previously Presented) A method of processing data comprising:  
receiving an interrupt indicating data from a local area network (LAN) has been stored in one of a plurality of buffers and is ready for processing;  
when a software buffer index points to a first buffer containing processed data, sequentially searching through said plurality of buffers containing data to determine whether there is a second buffer with unprocessed data; and  
if there is said second buffer with unprocessed data, synchronizing said software buffer index to a hardware buffer index by resetting said software buffer index to a next available buffer having processed data following said second buffer, and otherwise stopping said searching when each buffer of said plurality of buffers has been searched and a buffer with unprocessed data is not found.

11. (Original) The method of processing data as described in Claim 10, wherein said data from said LAN is a LAN packet.

12. (Previously Presented) The method of processing data as described in Claim 10, wherein a LAN driver performs said receiving, said searching and said synchronizing.

13. (Original) The method of processing data as described in Claim 10, further comprising:

determining if said first buffer contains processed data; and  
processing data in said first buffer if said data is unprocessed.

14. (Previously Presented) The method of processing data as described in Claim 10, further comprising:

when said software buffer index points to said first buffer containing processed data, using a value of said software buffer index corresponding to said first buffer as a reference value;

incrementing said software buffer index as each buffer of said plurality of buffers is searched, wherein when said software buffer index reaches one end of a range of possible values it is reset to the other end of said range; and

stopping said searching when said software buffer index reaches said reference value without finding a buffer in said plurality of buffers with unprocessed data.

15. (Original) The method of processing data as described in Claim 10, further comprising:

processing said unprocessed data in said second buffer.

16. (Previously Presented) A computer system comprising:

a processor; and

a computer readable memory coupled to said processor and containing program instructions that, when executed, implement a method of processing data, comprising:

when a software buffer index points to a first buffer containing processed data, synchronizing said software buffer index to a hardware buffer index by sequentially searching through a plurality of buffers containing data to determine whether there is a second buffer with unprocessed data; and

if there is said second buffer with unprocessed data, resetting said software buffer index to a next available buffer having processed data following said second buffer, and otherwise stopping said searching when each buffer of said plurality of buffers has been searched and a buffer with unprocessed data is not found.

17. (Previously Presented) The computer system as described in Claim 16, wherein said synchronizing further comprises:

synchronizing said hardware buffer index and said software buffer index in response to an interrupt indicating data has been stored in one of said plurality of buffers and is ready for processing.

18. (Previously Presented) The computer system as described in Claim 16, wherein said synchronizing further comprises:

ignoring a first interrupt indicating data has been stored in one of said plurality of buffers and is ready for processing when said software buffer index points to said first buffer containing processed data; and

synchronizing said hardware buffer index and said software buffer index in response to a second interrupt indicating data has been stored in one of said plurality of buffers and is ready for processing when said software buffer index points to said first buffer containing processed data for a second time.

19. (Original) The computer system as described in Claim 16, wherein said method further comprises:

determining if said first buffer contains processed data; and  
processing data in said first buffer if said data is unprocessed.

20. (Previously Presented) The computer system as described in Claim 16, wherein said synchronizing further comprises:

wrapping around to a start buffer after searching the end buffer in said plurality of buffers when sequentially searching through said plurality of buffers, said plurality of buffers sequentially beginning with a start buffer and ending with an end buffer.

21. (Previously Presented) The computer system as described in Claim 16, wherein said method further comprises:

when said software buffer index points to said first buffer containing processed data, using a value of said software buffer index corresponding to said first buffer as a reference value;

incrementing said software buffer index as each buffer of said plurality of buffers is searched, wherein when said software buffer index reaches one end of a range of possible values it is reset to the other end of said range; and

stopping said searching when said software buffer index reaches said reference value without finding a buffer in said plurality of buffers with unprocessed data.

22. (Original) The computer system as described in Claim 16, wherein each of said plurality of buffers is a local area network (LAN) buffer for storing LAN packets of data.

23. (Original) The computer system as described in Claim 22, wherein said software buffer index is a LAN software buffer index, and said hardware buffer index is a LAN hardware buffer index.

24. (Original) The computer system as described in Claim 16, wherein said method further comprises:

processing said unprocessed data in said second buffer.